DOD Best Practices Clearinghouse

David Castellano
OUSD(AT&L)/DS/SE/AS

Systems Engineering Forum
October 15, 2004

How Do We Encourage Broader Use of Best Practices?

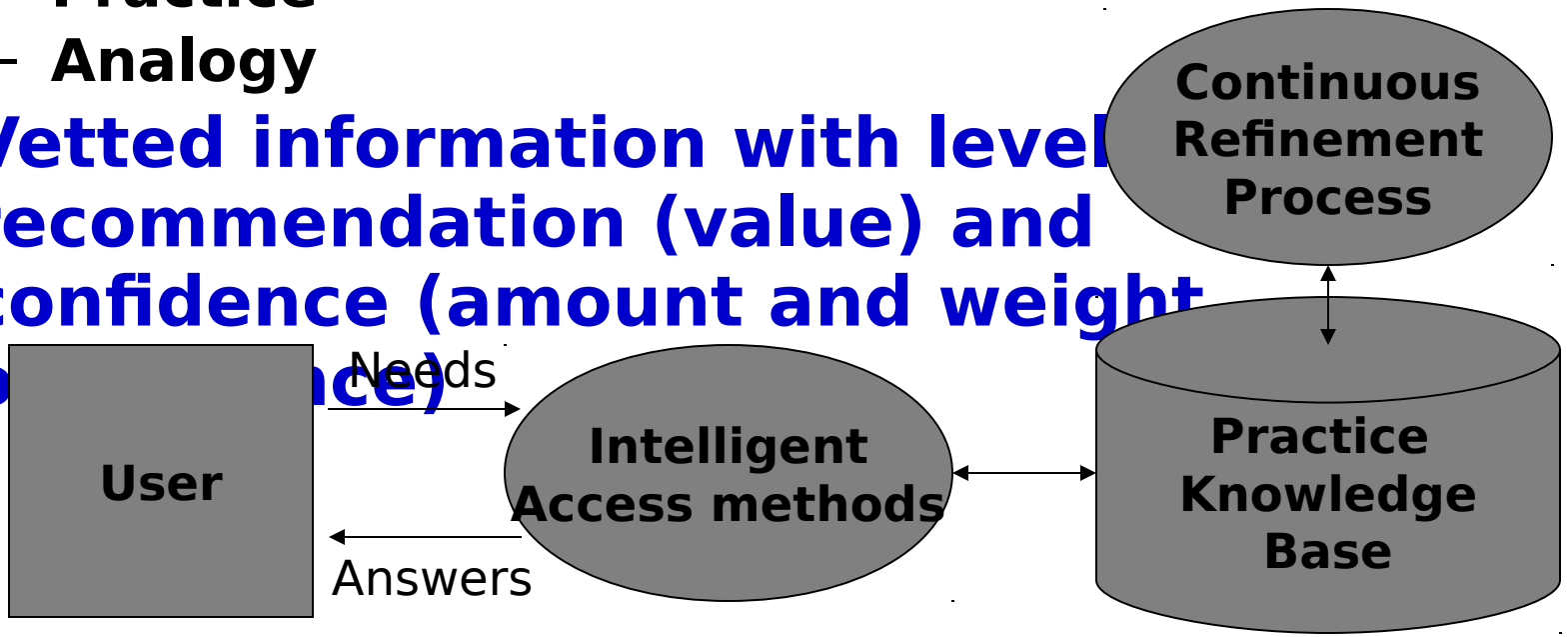
- **Section 804 directs DoD to establish a BP clearinghouse**
- **BPs recommended ad nauseam in the past**
  - **SEI, DSB, GAO, Congress**
- **Awareness of BPs is broad – Implementation poor**
- **Research identified a number of barriers**
  - **Too many lists**
  - **No basis for selection**
  - **No proof of effectiveness**
  - **No connection to risks**
  - **No clear success factors**
  - **No accurate costs/benefits**

ate
ntation
e

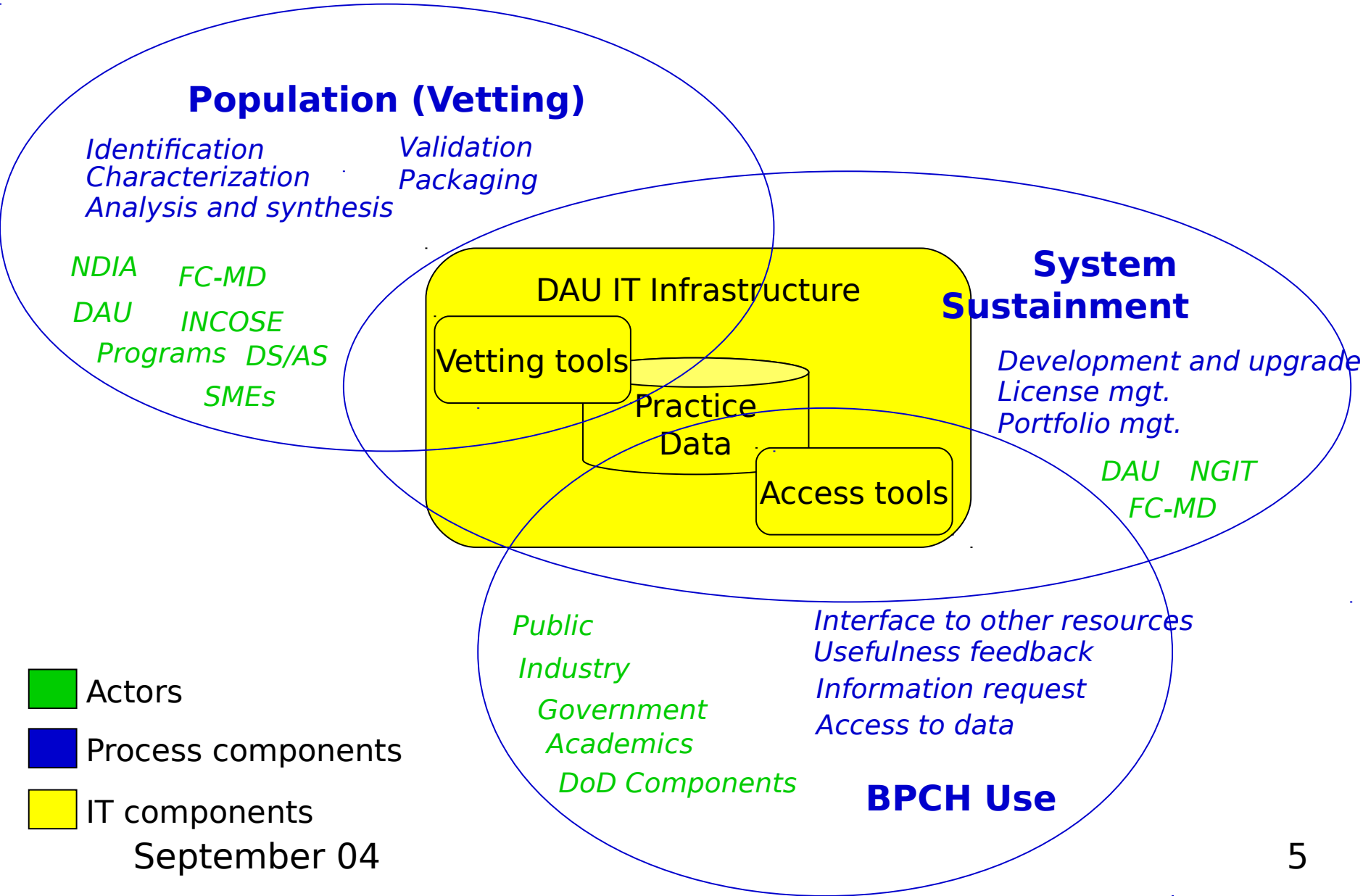
Who Uses the Clearinghouse?

The Concept - A Single Resource

- **Useful information tailored to the user context**
- **Support for practice selection based on multiple access vectors**
  - **Risk**
  - **Practice**
  - **Analogy**
- **Vetted information with level of recommendation (value) and confidence (amount and weight of evidence)**



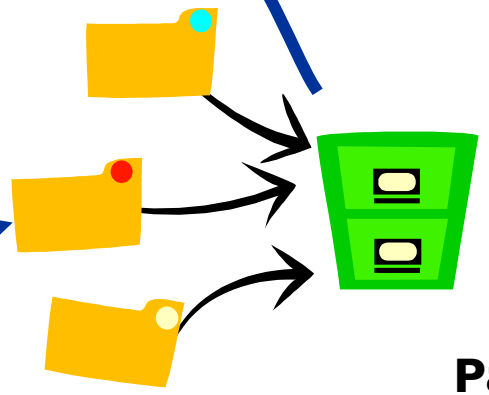
Clearinghouse Structure



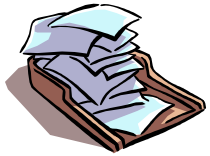
Best Practices Clearinghouse Process

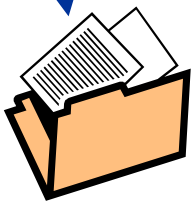
IN:
Studies, lessons learned, data

Practice Processors



OUT:
Packaged practice data with value and confidence indicators





Vetting team (NDIA, INCOSE, SEI, Academia)

Identification → **Characterization** → **Analysis & Synthesis** → **Vetting** → **Packaging & Dissemination**

Building an Empirical Knowledge Base: *Characterization, Analysis & Synthesis*

Models ensure consistency within the processes – support learning/improvement

Profile

	Maturity	The technical readiness of the practice (Using the NASA Technology Readiness Scale provided in instructions)	Immature (1-3)		**Adoptable (4-6)**	Mature (7-9)
		Indicate the impact magnitude and latency and list the items that are the cause of that incurred cost. Include percentages if possible. E.g. training 70%, infrastructure 30%.	Postive			
	Magnitude		Low		Medium	High
	Latency		Short		Medium	Long-term
	List items		<Text>		<Text>	<Text>
	n quality	Experience shows an immediate positive impact on quality, but it is not enough to eliminate inspections and formal reviews.	**Postive**			
	agnitude		**Low**		Medium	High
	Latency		**Short**		Medium	Long-term
	Comments		<Text>		<Text>	<Text>
	sch e	Indicate the impact magnitude and latency. *Tasks are completed faster.*	**Postive**			
	Magnitude		Low		**Medium**	High
	Latency		**Short**		Medium	Long-term
	Comments		<Text>		<Text>	<Text>
	Implementation	How much of a headache is it to deploy the practice. Include comments of what the difficulties were.	Very Easy		Somewhat Easy	Moderate
	Comments		<Text>		<Text>	<Text>
	nt Cost	*Experience shows that pair programming requires*	**Low**		*Medium*	*High*
	urden	ajor costs of the investment are	**Developer pays most**		Developer pays more	Evenly divided
	Requir	Additional training/expertise required (Hours of ining/Adjustment period in hours)	None		**Little (<8)**	Moderate (8-40)
		ence and observations show that in ent period varies from a few hours	Rarely		Sometimes	Often

Model for valuation of the maturity[1] of a practice

Attribute	Descriptive Value	Numerical Value
How long the practice has been around	Less than 1 year	1
	Less than 5 years	2
	More than 5 years	
Magnitude of problem to which the practice has been applied (Pick "best" value)	Unclear	
	Problem that took 40 hours (one person week worth of eff	
	or less per person t	
	Problem th (one p per	
	Pro (one per pers	
	Problem that (one person year wo per person to solve	

Model

Analysis & Synthesis

Skeptic Trace-back

Pair Programm pact on quali

Posi act

There is a positive impact on quality, but it is not eno imi

Several experts in the field argue that pair programming is a conti

are detected. A small scale experiment with students has shown a rea

Programming. Lessons Learned from several long-term industrial pilot proj

does help finding defects, however it doesn't detect all defects and formal inspections

programming.

Summary	Statement	
Pair Programming's *potential* of reducing defects: Several experts in the field argue that pair programming is a continuous review process and immediately defects are detected.	This practice provides the driver (the programmer at the keyboard) with feedback on his or her ideas and code. Together, the TDD and pair programming focus on continual quality assurance.	Will
	With pair programming, four eyeballs are better than two, and a huge number of defects are prevented right from the start.	Will
	When we finally made it to the first checkpoint application, we zoomed through QA with hardly a hitch. Everyone, myself included, was amazed that it didn't take weeks to debug, especially given that one of the trees had recently spent SIX WEEKS in QA hell. It was obvious that the pairs had dramatically reduced the defect rate.	oc lli
	Pair programming's shoulder-to-shoulder technique serves as a continual de , leading to most efficient defect removal rates.	
		Value:

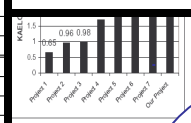
Summary

Valuation of empirical evidence

		Descriptive Value	Numerical Value
			1
	applied		1
the practice	r(s)		2
How the practice was applied	Unclear		1
	One small scale experiment		1
	One large scale experiment		2
	One industrial pilot project		2
	eriments		
			5

Model

Characterization



Collect source data

Sources

Useful Information

- **Project or context-specific practice selection or suggestion support**
- **Practice information at various levels of detail**
- **Experience reports, lessons learned, expert**

Experience data

Case Study #24	
Best practice	Formal inspections
Source	"Report on the Loss of the Mars Climate Orb Mission", [J PL D-18441, J PL Special Review Board, Nov. 11, 1999]

Theory/Expectation

The use of software inspections will ensure a high level of system qual

Lesson Learned

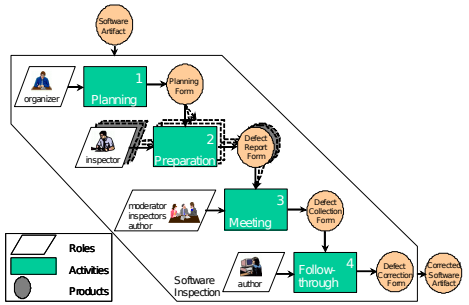
Attention must be paid that inspections are practiced correctly, with appropriate formality, to ensure defect removal benefits.

What happened

Breakdown in the use of inspections:
- Contrary to typical practice, there was not a requirement for a navigation (end-user) representative to be present at any of the walkthroughs or the acceptance test.
- The Sm_forces software program was misclassified as non-mission critical, which reduced the number of reviews done on the software compared to mission critical software.

Implementation data/ guidance

Inspection process overview



Phase 1: Planning

Inspectors should have vested interests in work product

Inspectors should invest no more than 15% of their time in inspections (don't overwork good inspectors!)

…

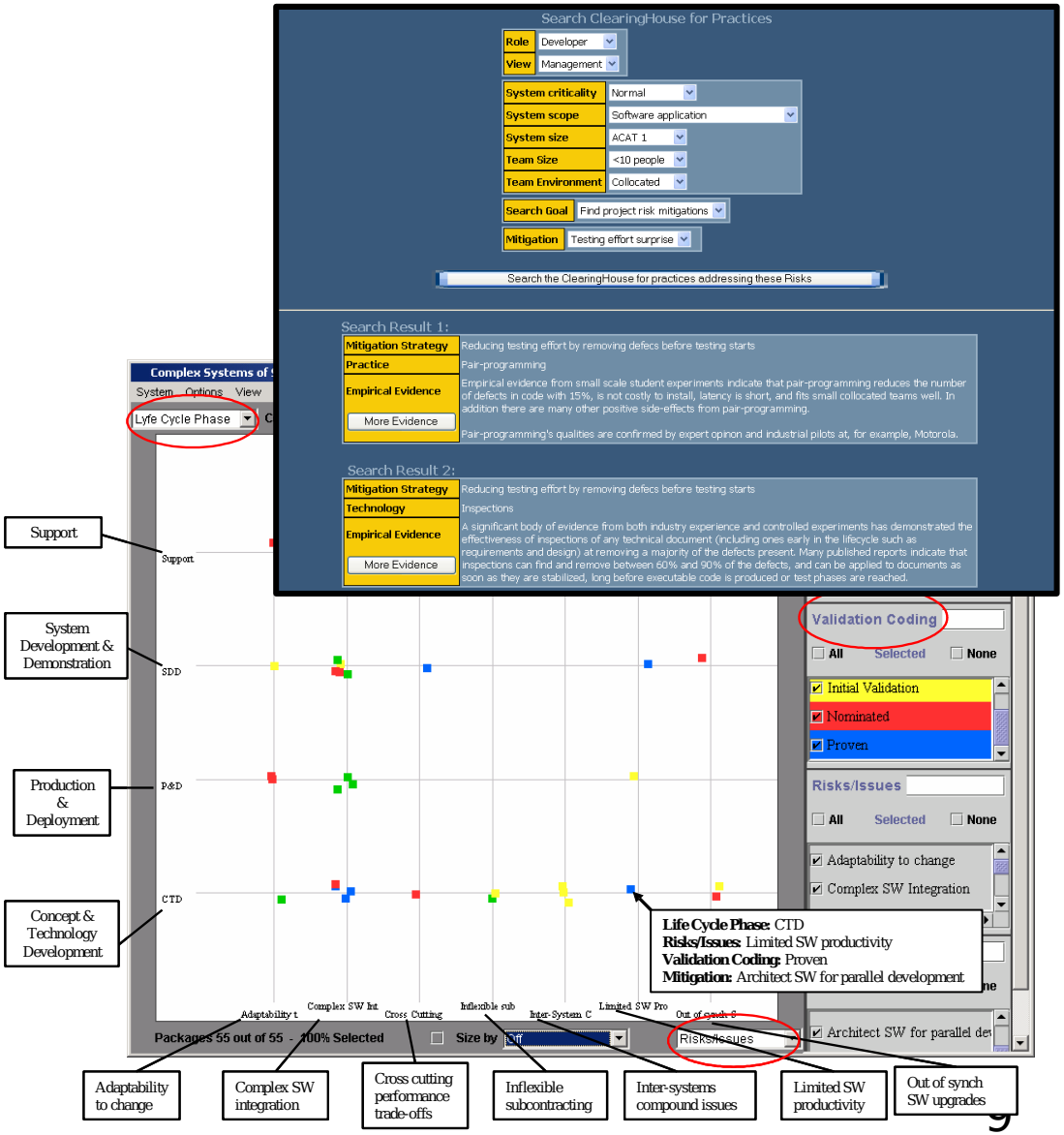
Phase 2: Preparation

Inspectors should spend at least as much time in preparing as is required for the inspection meeting.

Provide adequate lead time for inspectors to perform preparation (3 - 5 work days)

Intelligent Front-end

- **Focused on relevant information**
- **Presents information from user perspective**
- **Data accessible by multiple, tailorable views**
- **Utilizes push and pull methods**
- **Provides wizards, search tools, indexes, categories**
- **Relies on evolving ontology and cutting edge expert system technology**



BPCH Status

- **Designated as the single DoD source for validated acquisition practices**
  - **Added to the Acquisition Domain IT portfolio**
  - **DAU sees major role within their integrated knowledge management and educational infrastructure**
- **Development team in place**
  - **Fraunhofer Center, NGIT developers**
  - **DAU providing strong support to ensure integration**
  - **NDIA/INCOSE task group formed**
- **Key Events**
  - **October 25-28 – NDIA SE Conference (Dallas)**
    - ***Presentation and concept video***
    - ***Questionnaire/feedback instrument***
  - **November 15 – PEO/SYSCON (Ft. Belvoir)**
    - ***Included in Mr. Wynne's/Dr. Lamartin's talk***
    - ***One pager for conference notebook***
  - **April 18-21, 2005 – System and Software Technology Conference (Salt Lake City)**
    - ***Presentation and BPCH Prototype***
      - **Prototype reviews in January and March '05**